[redacted]

IN THE CLAIMS

Please amend the claims to read as provided below.

1. (Currently Amended) A system that provides for remote password authentication, comprising:

a client;

a plurality of authentication servers;

a network interconnecting the client and the plurality of authentication servers; and

a memory, coupled to the client, the memory maintaining instructions that when executed by the client, cause the client to receive a password, transmit a unique random value y_i to each of the servers, derive a group element (P) from the password, send a blinded password value (P^x) to the servers, receive blinded key shares (P^{xy_i}) from the servers, unblind and combine the ~~blinded~~ key shares to ~~erate~~ create a master key (K_m), and decrypt encrypted private data using the master key (K_m)

2. (Previously Presented) The system recited in Claim 1 wherein the instructions further cause the client to validate the master key (K_m).

3. (Currently Amended) The system recited in Claim ~~1~~ 2 wherein the instructions further cause the client to decrypt encrypted private data using the validated master key (K_m).

4. CANCEL

5. (Previously Presented) The system recited in Claim 2 wherein the instructions further cause the client to send proof of the validated master key (Km) and each blinded password value (P^x) to the servers.

6. (Currently Amended) A method that provides for remote password authentication using a system including a client, a plurality of authentication servers, and

a network interconnecting the client and the plurality of authentication servers, the method comprising the steps of:

receiving a password;

deriving group elements (P) from the password;

sending a blinded password value (P^x) to the servers;

receiving blinded key shares (P^{xy_i}) from the servers;

unblinding and combining the ~~blinded~~ key shares to create a master key (K_m); and

decrypting encrypted private data using the master key (K_m).

7. (Original) The method recited in Claim 6 further comprising the step of validating the master key (K_m).

8. CANCEL

9. (Original) The method recited in Claim 7 further comprising the step of decrypting encrypted private data using the validated master key (K_m).

10. (Original) The method recited in Claim 7 further comprising the step of sending proof of the validated master key (K_m) and each blinded password value (P^x) to the servers.

11. (Previously Presented) A computer program embodied on a computer-readable medium for enabling remote password authentication in a multiple-server system including a client, a plurality of authentication servers, and a network interconnecting the client and the plurality of authentication servers, the computer program comprising:

a code segment that enters a password;

a data storage area that contains a unique random value y_i on each of the servers,

a code segment that derives a group element (P) from the password;

a code segment that sends blinded password value (P^x) to the servers;

a code segment that provided for receiving blinded key shares (P^{xy_i}) from the servers;

a code segment that unblinds and combines the shares to create a master key (K_m); and

a code segment that decrypts encrypted private data on the client computer using the master key (K_m).

12. (Original) The computer program recited in Claim 11 further comprising a code segment that validates the master key (K_m).

13. CANCEL

14. (Original) The computer program recited in Claim 12 further comprising a code segment that decrypts encrypted private data using the validated master key (K_m).

15. (Original) The computer program recited in Claim 12 further comprising a code segment that sends proof of the validated master key (K_m) and the blinded password value (P^x) to the servers.

16. (Previously Presented) The system recited in Claim 1 wherein the authentication servers include a memory for maintaining instructions which, when executed by the authentication servers, cause the authentication servers to:

maintain a count of bad login attempts, the number of recent amplifications, a list of recent P^x password amplification request values, and a list of timestamps associated with the list of recent password amplification request values on the server;

receives a blinded password (P^x) request

records the blinded password in a short-term list

checks a user account to see if it is locked;

creates a blinded key share (P^{xy_i}) in response to the blinded password request; and

sends the blinded key share to the client if it is unlocked.

17. (Previously Presented) The system recited in Claim 16 wherein the instructions further cause the authentication servers to:

records a timestamp value to note the time that the request was received;

periodically checks for stale requests which are determined when the difference between any timestamp value and the current time becomes greater than a specific period of time;

deletes corresponding password amplification request values and timestamps; and

increments the count of bad attempts.

18. (Previously Presented) The system recited in Claim 16 wherein, when a successful login occurs, the instructions further cause the authentication servers to:

sends a value of Q_A, equal to the password raised to a random power, along with any prior values for Q_A from earlier runs in the same login session, to each server in an encrypted message; and

authenticate the encrypted message using the master key K_m.

19. (Previously Presented) The method recited in Claim 6 further comprising the steps of:

maintaining a count of bad login attempts, the number of recent amplifications, a list of recent P^x password amplification request values, and a list of timestamps associated with the list of recent password amplification request values on the server;

receiving a blinded password (P^x) request

recording the blinded password in a short-term list

checking a user account to see if it is locked;

creating a blinded key share (P^{xy_i}) in response to the blinded password request; and

sending the blinded key share to the client if it is unlocked.

20. (Previously Presented) The method recited in Claim 19 further comprising the steps of:

recording a timestamp value to note the time that the request was received;

periodically checking for stale requests which are determined when the difference between any timestamp value and the current time becomes greater than a specific period of time;

checking corresponding password amplification request values and timestamps; and

incrementing the count of bad attempts.

21. (Previously Presented) The method recited in Claim 19 further comprising the steps of

sending the value of Q_A, equal to the password raised to a random power, along with any prior values for Q_A from earlier runs in the same login session, to each server in an encrypted message; and

authenticating the encrypted message using the master key K_m.

22. (Previously Presented) The computer program recited in Claim 11 further comprising a code segment that:

maintains a count of bad login attempts, the number of recent amplifications, a list of recent P^x password amplification request values, and a list of timestamps associated with the list of recent password amplification request values on the server;

receives a blinded password (P^x) request

records the blinded password in a short-term suspect list

checks a user account to see if such account is locked;

creates a blinded key share (P^{xy_i}) if the user account is unlocked; and

sends the blinded key share to the client.

23. (Original) The computer program recited in Claim 22 further comprising a code segment that:

records a timestamp value to note the time that the request was received;

periodically checks for stale requests which are determined when the difference between any timestamp value and the current time becomes greater than a specific period of time;.

deletes corresponding password amplification request values and timestamps; and

increments the count of bad attempts.

24. (Original) The computer program recited in Claim 22 further comprising a code segment that:

sends the value of Q_A, equal to the password raised to a random power, along with any prior values for Q_A from earlier runs in the same login session, to each server in an encrypted message; and

authenticates this message using the master key K_m.